# 1 什么是反射

在java当中,一个类，或者对象，都是可以找到其对应的属性和方法

# 2 反射类的例子

1 class

**package** com.company.reflect;  
  
*/\*\*  
 \** ***@Auther:*** *qlong2  
 \** ***@Date:*** *2019/2/19 16:52  
 \** ***@Description:*** *\*/***public class** Robot {  
 **private** String **name**;  
  
 **public void** sayHi(String msg) {  
 System.***out***.println(msg +**" "** +**name**);  
 }  
  
 **private** String sayHello(String user){  
 **return "hello "** + user;  
 }  
}

main

**package** com.company.reflect;  
  
**import** java.lang.reflect.InvocationTargetException;  
**import** java.lang.reflect.Method;  
  
*/\*\*  
 \** ***@Auther:*** *qlong2  
 \** ***@Date:*** *2019/2/19 16:57  
 \** ***@Description:*** *实现一个反射的例子  
 \** ***@Description;*** *找到类的属性和方法，并执行  
 \*/***public class** ReflectSample {  
 **public static void** main(String[] args) **throws** ClassNotFoundException, IllegalAccessException, InstantiationException, NoSuchMethodException, InvocationTargetException {  
 *//通过Class.forName得到Class* Class rc = Class.*forName*(**"com.company.reflect.Robot"**);  
 System.***out***.println(**"class name is "** + rc.getName());  
 Robot robot = (Robot) rc.newInstance();  
 *// 得到Robot的private方法 getDeclaredMethod 只能得到该类的方法，父类不行* Method method = rc.getDeclaredMethod(**"sayHello"**,String.**class**);  
 method.setAccessible(**true**);  
 Object o = method.invoke(robot,**"Kyrie"**);  
 System.***out***.println(**"the sayHello result "** + o);  
 *//得到方法，必须要传参数，不传参数会报 NosuchMethodException* Method publicMethod = rc.getMethod(**"sayHi"**,String.**class**);  
 Object sayHiObject =publicMethod.invoke(robot,**"Welcome"**);  
 System.***out***.println(**" the sayHi result "** + sayHiObject);  
 */\*Method publicMethod = rc.getDeclaredMethod("sayHi",String.class);  
 Object sayHiObject = publicMethod.invoke(robot,"Welcome");  
 System.out.println(sayHiObject);\*/* }  
}